

Envisia Physically Knowledgeable Synthesis Product Notes

**Product Version 4.0.8
May 2001**



© 1999-2000 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 1-800-862-4522.

All other trademarks are the property of their respective holders.

Restricted Print Permission: This publication is protected by copyright and any unauthorized use of this publication may violate copyright, trademark, and other laws. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This statement grants you permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used solely for personal, informational, and noncommercial purposes;
2. The publication may not be modified in any way;
3. Any copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement; and
4. Cadence reserves the right to revoke this authorization at any time, and any such use shall be discontinued immediately upon written notice from Cadence.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

| | |
|--|----|
| <u>Product Notes for Cadence[®] PKS</u> | 5 |
| <u>Low Power Synthesis Option of Cadence PKS</u> | 7 |
| <u>New in Version 4.0.8</u> | 7 |
| <u>CTPKS</u> | 7 |
| <u>New in Version 4.0.8</u> | 7 |
| <u>Previously Included CTPKS Features in Versions 4.0 to 4.0.7</u> | 9 |
| <u>Test Synthesis Enhancements</u> | 11 |
| <u>New in Version 4.0.8</u> | 11 |
| <u>LEF Enhancements</u> | 12 |
| <u>New in Version 4.0.8</u> | 12 |
| <u>Previously Included LEF Enhancements in Versions 4.0 to 4.0.7</u> | 12 |
| <u>DEF Enhancements</u> | 12 |
| <u>New in Version 4.0.8</u> | 12 |
| <u>Previously Included DEF Enhancements in Versions 4.0 to 4.0.7</u> | 12 |
| <u>ADB Enhancements</u> | 13 |
| <u>New in Version 4.0.8</u> | 13 |
| <u>Previously Included ADB Enhancements in Versions 4.0 to 4.0.7</u> | 13 |
| <u>QP/WR Enhancements</u> | 13 |
| <u>New in Version 4.0.8</u> | 13 |
| <u>PKS GUI Enhancements</u> | 13 |
| <u>New in Version 4.0.8</u> | 13 |
| <u>Previously Included GUI Enhancements in Versions 4.0 to 4.0.7</u> | 13 |
| <u>Shield Net Support</u> | 15 |
| <u>New in Version 4.0.8</u> | 15 |
| <u>Handling Power Rails</u> | 15 |
| <u>New in Version 4.0.8</u> | 15 |
| <u>Using Non-TLF 4.3 Libraries</u> | 16 |
| <u>New in Version 4.0.8</u> | 16 |
| <u>Improved Repeater Insertion Code</u> | 17 |
| <u>New in Version 4.0.8</u> | 17 |
| <u>Previously Included Repeater Insertion Code Features in Versions 4.0 to 4.0.7</u> | 17 |

Product Notes for Cadence PKS

| | |
|--|----|
| <u>Improved Congestion Analysis</u> | 17 |
| <u>New in Version 4.0.8</u> | 17 |
| <u>Previously Included Congestion Analysis Features in Versions 4.0 to 4.0.7</u> | 17 |
| <u>Handling of High Fanout Nets During PKS Optimization</u> | 18 |
| <u>New in Version 4.0.8</u> | 18 |
| <u>Passing pin_cap Information from the Synthesis Library to the WDB file</u> | 18 |
| <u>New in Version 4.0.8</u> | 18 |
| <u>Previously Included pin_cap Features in Versions 4.0 to 4.0.7</u> | 19 |
| <u>New Commands</u> | 19 |
| <u>New in Version 4.0.8</u> | 19 |
| <u>Previously Included PKS Commands in Versions 4.0 to 4.0.7</u> | 19 |
| <u>Global Commands</u> | 19 |
| <u>New in Version 4.0.8</u> | 19 |
| <u>Previously Included Global Commands in Versions 4.0 to 4.0.7</u> | 19 |
| <u>Updated Commands</u> | 20 |
| <u>New in Version 4.0.8</u> | 20 |
| <u>Previously Included Updated Commands In Versions 4.0 to 4.0.7</u> | 20 |
| <u>Fix List for Version 4.0.8</u> | 23 |
| <u>Fix List for Version 4.0.7</u> | 24 |
| <u>Fix List for Version 4.0.6</u> | 24 |
| <u>Fix List for Version 4.0.5</u> | 26 |
| <u>Fix List for Version 4.0.4</u> | 26 |

Product Notes for Cadence[®] PKS

May 2001

This product note highlights what is new in this release of Cadence[®] physically knowledgeable synthesis (PKS).

Note: Refer to the *Product Notes for Ambit[®] BuildGates[®] Synthesis* for additional features and enhancements that also apply to PKS.

This release includes the following features and enhancements.

- [Low Power Synthesis Option of Cadence PKS](#) on page 7
- [CTPKS](#) on page 7
- [Test Synthesis Enhancements](#) on page 11
- [LEF Enhancements](#) on page 12
- [DEF Enhancements](#) on page 12
- [ADB Enhancements](#) on page 13
- [QP/WR Enhancements](#) on page 13
- [PKS GUI Enhancements](#) on page 13
- [Shield Net Support](#) on page 15
- [Handling Power Rails](#) on page 15
- [Using Non-TLF 4.3 Libraries](#) on page 16
- [Improved Repeater Insertion Code](#) on page 17
- [Improved Congestion Analysis](#) on page 17
- [Handling of High Fanout Nets During PKS Optimization](#) on page 18
- [Passing pin_cap Information from the Synthesis Library to the WDB file](#) on page 18
- [New Commands](#) on page 19
- [Global Commands](#) on page 19

Product Notes for Cadence PKS May 2001

- Updated Commands on page 20
- Fix List for Version 4.0.8 on page 23
- Fix List for Version 4.0.7 on page 24
- Fix List for Version 4.0.6 on page 24
- Fix List for Version 4.0.5 on page 26
- Fix List for Version 4.0.4 on page 26

Low Power Synthesis Option of Cadence PKS

New in Version 4.0.8

If you have the Low Power Synthesis (LPS) option of Cadence PKS, you can run all LPS commands and options with PKS, with these post-placement exceptions:

- You can not run the `do_optimize` command using the `-power` and `-pks` options at the same time. Each option works with this command on its own.
- The `do_xform_optimize_power` command does not have a `-pks` option to allow it to run within Cadence PKS.

Note: Commands that you *can* run after placement are the power analysis commands—`get_power` and `report_power`—and the power-conscious timing optimization command `do_xform_optimize_slack -power`.

See the Command Reference for Ambit BuildGates Synthesis and Cadence PKS for details on all LPS commands.

CTPKS

CTPKS runs inside PKS to generate clock trees. Its purpose is to find the best topology that meets your constraints. It finds both the best tree structure and the best location (x,y) for all components in the clock tree.

New in Version 4.0.8

Signal Integrity (Wire Self Heat) Support

- CTPKS calculates and reports the number of current violations in a clock tree.
- CTPKS works to minimize the number of current violations when building a clock tree.
- CTPKS propagates `DEF SHIELDNET` attributes as well as net attributes to all tree nets if they are present on the root. This includes: `NONDEFAULTRULES`, Layer Usage Table, R and C LEF multipliers .

Product Notes for Cadence PKS May 2001

Clock Gating Components

Clock gating components can be automatically moved when building a tree in order to improve performance.

If the original placement of a gating component is not suitable, it can be moved closer to its own leaves and closer to the main root of the tree.

Low Power Consideration

Low power criterium can be selected when building a clock tree. Note that this requires that you have the license for the LPS option for Ambit BuildGates synthesis and Cadence physically knowledgeable synthesis (PKS).

After CTPKS has performed the different trials while building a clock tree, it keeps the one that has the lowest power consumption for the clock tree and flip-flops, provided that the clock tree constraints are respected.

Arbitrary Insertion on Leaf Pins

An arbitrary insertion delay can be specified on any leaf pin.

A value, modeling the clock insertion delay of the logic downstream from a particular pin, may be set on this pin in order for it be taken into account in balancing the tree.

Usability Improvements

- Clock tree constraints are automatically saved in the `.adb` file when running the `write_adb` command.
- A new `reset_clock_tree_constraints` command is available.
- A pin with attribute `ct_excluded true` is now allowed on the inverting path from the (sub-)tree root.
- Constraints for delay, skew, and transition are available for physical trees (non-clock signals).

Constraints values can now be specified. Previous versions used the default values for minimum delay, maximum delay and maximum skew. Now, both edges are considered for delay or skew calculation.

- An additional `-fast` switch for `do_build_clock_tree` and `do_build_physical_tree` enables faster completion of a job.

Product Notes for Cadence PKS May 2001

- Runtime for the command `do_build_physical_tree` has been improved, on average, by a third.
- The report header for the `report_clock_tree` and `report_clock_tree_violations` commands now contain the following:
 - Clock tree area
 - Number of inverters
 - Buffers
 - Number of violations
- The `-no_gated` option of the `do_build_clock_tree` command lets you build a balanced tree to the inputs of non-buffer, non-inverter cells.

Refer to the *PKS User Guide* for more information about CTPKS.

Previously Included CTPKS Features in Versions 4.0 to 4.0.7

- [Clock Tree Generation](#) on page 9
- [Clocking Structure](#) on page 10
- [Buffering Long Nets](#) on page 10
- [Delay Calculation](#) on page 10

Clock Tree Generation

- CTPKS constructs buffer trees for any number of specified clocks in a design.
- Clock trees are built one by one.
- CTPKS explores different tree structures. For each structure, CTPKS places the new buffers/inverters, and it groups the clock leaf pins (CLOCK input pins) into clusters so that pin and wire loads are balanced.
- CTPKS computes an ideal (x and y) location, and the PKS infrastructure provides the closest legal (x and y) location.
- PKS timing verification runs on the newly created clock tree with parasitics extracted from the (x and y) location of each buffer/inverter. If the performances of the newly created tree are better than that of the previous tree, it is kept; otherwise, it is rejected.

Product Notes for Cadence PKS May 2001

Clocking Structure

- CTPKS considers the clock tree to be the entire sub-circuit from a given root to the component input pins that have been declared as `CLOCK` pins in the TLF or the ALF files (default operation).
- CTPKS removes any existing buffers or inverters that are traced through. If the command is set on the instance, this does not occur. If you enter the `set_dont_modify` command, the specified object is kept.
- A clock tree can have multiple levels of gating, as well as gating components with multiple outputs.
- A clock tree can have blocks with embedded clock trees. The timing for the embedded clock tree is specified using the `INSERTION_DELAY` statement in TLF 3.1.
- CTPKS decomposes the clock tree into subtrees. Each subtree has only one root and contains only buffers and inverters that CTPKS is allowed to delete. Those buffers and inverters that CTPKS is not allowed to delete (those marked `DONT_TOUCH` or `DONT_MOVE`) and gated cells are treated as subtree delimiters.
- CTPKS categorizes the leaf pins of each subtree into two types:
 - Pins on non-inverting paths from the subtree root
 - Pins on inverting paths from the subtree root

CTPKS builds a non-inverting tree from the subtree root to the pins on non-inverting paths and an inverting tree from the subtree root to the pins on inverting paths.

Buffering Long Nets

CTPKS can create balanced buffer trees for long nets, such as `set` or `reset` nets. Using this model, you can ask for a `no_gated` tree and the tree stops at any gated components. The skew is not usually a constraint (refer to [do_build_physical_tree](#) and the `no_gated` option of the [report_clock_tree](#) command).

Delay Calculation

Support for delay calculation includes the following:

Parasitics

CTPKS performs its clock tree building based on the parasitics estimations returned by PKS placement and steiner tree or half-perimeter evaluations.

Product Notes for Cadence PKS May 2001

Slew Propagation

CTPKS uses the slew propagation model that is set.

PVT Support

Only one case of PVT (Process, Voltage, Temperature) can be handled at any given time. Therefore the clock tree will be built for one particular PVT. In the flow, the user will have to check the clock trees for other PVTs afterwards.

Commands

The following commands are included with CTPKS (listed in functional order):

- set clock tree constraints
- get clock tree constraints
- get clock tree objects
- set attribute
- do build clock tree
- report clock tree
- report clock tree violations
- do build physical tree

Test Synthesis Enhancements

New in Version 4.0.8

- Test synthesis of clocked-scan, clocked-LSSD and aux-clocked LSSD scan styles.
- Insertion of data lockup latches between user-specified compatible clock domains.
- PKS reordering of scan chain segments separated by data-lockup latches.
- Multiple scan enables in single clock domain/multiple chains or multi-clock domain designs.
- Shared scan-data in with functional input ports.

Product Notes for Cadence PKS May 2001

- Shared scan-data out with functional output ports through insertion of a multiplexor.
- Support for test mode, scan mode and scan-data to pins of technology components or lower level module ports (hierarchical pins).
- Support for internal clock domain points as separate DFT scan chain domains.

Refer to the *Test Synthesis for Ambit BuildGates Synthesis and Cadence PKS* for more information.

LEF Enhancements

New in Version 4.0.8

There are no new LEF enhancements in version 4.0.8.

Previously Included LEF Enhancements in Versions 4.0 to 4.0.7

- Native support for LEF constructs on via rules, spacing rules, and nondefault rules.
- LEF reader upgrades to handle syntax supported by the LEF v5.4 parser.

DEF Enhancements

New in Version 4.0.8

There are no new DEF enhancements in version 4.0.8.

Previously Included DEF Enhancements in Versions 4.0 to 4.0.7

- Read rows and tracks
- Handle net properties

Note: If rows are read in through DEF, the rows become the master template for PKS. PKS assumes these rows are available for placement. In this case, floorplan parameters are ignored. However, if placement areas are defined, then only DEF rows that lie within the placement area are used for placement.

ADB Enhancements

New in Version 4.0.8

There are no new ADB enhancements in version 4.0.8.

Previously Included ADB Enhancements in Versions 4.0 to 4.0.7

- The information saved in the ADB now contains information about new objects (physical instances, special nets, etc.) as well as properties and attributes for different objects.

QP/WR Enhancements

New in Version 4.0.8

- Several new options to control QP/WR have been added to the `do_place` and `do_route` commands. Refer to the *Command Reference for Ambit[®] BuildGates[®] Synthesis and Cadence[®] PKS* for more information on these options.
- Support for instance weight.
The instance can be set with a weight. This is requested by CTPKS. QP ECO will recognize weighted instances and keep them from moving if at all possible.
- Nets can be labeled with a source property.
Nets can now have a source property, such as netlist, dist, user, test, or timing.

PKS GUI Enhancements

New in Version 4.0.8

There are no new GUI enhancements in version 4.0.8.

Previously Included GUI Enhancements in Versions 4.0 to 4.0.7

- A mini-zoom window is available in the cluster browser window.

Product Notes for Cadence PKS May 2001

- By default, the object selection is a single mouse click. Multiple selection is performed using the shift-mouse-2 button sequence.
- Objects displayed, such as airline, steiner tree, and critical path, are now persistent with zoom operations. A new button cancels their display.
- Access to a cross selection of objects is available within the cluster browser.
- Arrow keys are bounded to pan operations.
- A new cluster hierarchy display is available within the cluster browser.
- Objects are selectively displayable (outlined/filled) for instances, macros and cluster pins.
- Persistent congestion display. Congestion display is compatible with zoom operations.
- A dynamic bin congestion report is included in the status window.
- Editable PKS objects, that include the following:
 - Move/rotate/mirror instances
 - Move cluster pins
 - Create/delete placement areas
 - Create/delete blockages
- Multi-object selection alignment (left/right/top/bottom).
- Blockages are selectively displayable using display.
- New layer selection dialog.
- Cluster row selective display.
- New menu command for placement spreading.
- New timing report window for selected critical paths in the cluster browser. A double spreadsheet table reports the timing path information.
- Selective instance timing information report.
- Verilog and VHDL escaped names are supported.

Shield Net Support

New in Version 4.0.8

If the input DEF to PKS contains net attributes related to shield nets and shield routes, PKS stores the `SHIELDNET` and `SHIELDROUTE` attributes of the net in the database.

All information related to shield handling is passed downstream to `do_route` or `do_place` and is written out in the DEF using `write_def`.

Handling Power Rails

New in Version 4.0.8

During PKS optimization, PKS makes estimates on the amount of resources used by supply rails in the floorplan. Supply rails are the power lines that connect the standard cells and are created by the `-followpins` command in Silicon Ensemble™ Place-and-Route.

If you provide a DEF file containing these power rails, PKS uses the rails in its congestion analysis and global routing. If these power rails are not provided in the DEF, PKS estimates congestion due to the rails during optimization and new rails are generated during the `do_route` step.

If you provide the power rails through a cover macro which is instantiated in your design, PKS cannot detect that the power rails have been provided and will create its own power rails as described above. To prevent this from happening, you must set the following global:

```
set_global estimate_supply_rail_congestion false
```

Then when you run `do_route`, use the `-no_followpins` option:

```
do_route -no_followpins
```

If a design contains feed through supply rails, the pins are always deleted at the beginning of `do_place`, and re-generated at the beginning of `do_route`.

New commands and options related to supply rail handling:

- The `do_route` command has a `-followpins` option which causes existing followpins to be removed and new ones to be generated.
- The `remove_supply_rails_on_rows` command deletes existing followpins.

Product Notes for Cadence PKS May 2001

- The `generate_supply_rails_on_rows` command generates new followpins.

RC Characterization

The following definitions are allowed:

- Multiple layer tables (`set_layer_usages_table`)
- Scale all values in technology LEF via TCL (`set_lef_multiplier`)
- Non-default rules (defined in LEF)

You can assign different combinations of these definitions to influence the steiner RC calculation by issuing the `set_net_physical_attribute` command. By default, two categories are considered:

- All nets synthesized by CTPKS (that have the `+use_clock` attribute in DEF)
- All other (non-clock-tree) nets

You can also make more specific assignments on a per-net basis. However, the 4.0.8 PKS release does not honor those assignments during optimization unless the nets are also marked as `dont_modify`.

Using Non-TLF 4.3 Libraries

New in Version 4.0.8

There have been a number of enhancements to the 4.0 timing engine which require more information than is available in TLF 4.3. If you still want to use TLF 3.x or TLF 4.1, then set the following global:

```
set_global auto_slew_and_delay_degradation false
```

After you issue the above command, do not try to read `rspf`, `dspf`, or `spef` into PKS to do delay calculation because your numbers will be wrong. TLF 4.3 has the slew measurement points (20%/80% or 10%/90%), and these functions use those values for better timing results.

It is recommended that you use `syn2tlf` version 4.3 to generate new TLF models.

Improved Repeater Insertion Code

New in Version 4.0.8

There are no new repeater insertion code features in version 4.0.8

Previously Included Repeater Insertion Code Features in Versions 4.0 to 4.0.7

Improvements were made to determine the placement location for the repeater along the steiner route, rather than at the center of the two pins (a repeater here is defined as a buffer on a 2-pin net). This is important for designs with blockages.

Improved Congestion Analysis

New in Version 4.0.8

There are no new congestion analysis features in version 4.0.8

Previously Included Congestion Analysis Features in Versions 4.0 to 4.0.7

Routing congestion affects the placement adjustment steps within PKS logic optimization. Routing congestion analysis has been improved and is more accurate. Changes to congestion analysis have been done to correlate PKS steiner-based congestion analysis better with wroute congestion analysis. The PKS congestion grid map size is determined using the g-cell grid size obtained from wroute. With these improvements, optimization does a better job of accounting for smaller congestion hot spots.

Handling of High Fanout Nets During PKS Optimization

New in Version 4.0.8

Cadence PKS now has more efficient handling of large fanout nets during PKS optimization.

The global, `large_fanout_size`, controls the number of fanouts at and beyond which a net is considered to have large fanouts. The default of the global is 200—the same number at and beyond which `do_place` considers a net to have large fanouts that are to be ignored during placement.

During pre-placement optimization, nets which have a number of fanouts greater than or equal to that specified by the `large_fanout_size` global are ignored during DRV fixing. However, if there are timing violations on these nets, the optimizer attempts to fix them.

After placement, these `large_fanout_size` nets are buffered with a fast physical buffer tree algorithm similar to CTPKS. For any nets that this physical buffer tree algorithm fails to buffer, one of the following actions is taken:

- If the net has a number of fanouts greater than or equal to 1000, it is marked as a large-fanout net and is ignored for subsequent DRV fixing or DRV calculations. If there are timing violations on these nets, the optimizer attempts to fix these timing violations. Because it can take the optimizer a long time to fix the violations, which hurts overall QOR, make sure that large-fanout nets do not have unnecessary or inadvertent timing constraints.
- If the net has numbers of fanouts less than 1000, it is buffered by the regular DRV fixing algorithm, which only considers reducing the number of fanouts without calculating and trying to fix other DRVs. It does this until this number gets below the `large_fanout_size` and becomes more manageable.

Note: To aid the review of nets which have fanouts that exceed a given threshold, use `report_net -hier -min_fanout n -summary`, where `n` is the fanout threshold you want considered.

Passing `pin_cap` Information from the Synthesis Library to the WDB file

New in Version 4.0.8

There are no new `pin_cap` features in version 4.0.8

Previously Included `pin_cap` Features in Versions 4.0 to 4.0.7

Originally, `pin_cap` information in the WDB file was derived from the LEF file. In the 4.0.x releases, `pin_cap` information that is saved in the WDB file is derived from the synthesis library.

New Commands

New in Version 4.0.8

There are no new PKS commands in version 4.0.8

Previously Included PKS Commands in Versions 4.0 to 4.0.7

- `do generate estcap`
- `get current congestion`
- `report supply rails on rows`
- `set power stripe spec`
- `set steiner mode`
- `set supply rails on rows`

Note: Refer to the *Command Reference for Ambit[®] BuildGates[®] Synthesis and Cadence[®] PKS* for more information about PKS commands.

Global Commands

New in Version 4.0.8

There are no new global PKS commands in version 4.0.8

Previously Included Global Commands in Versions 4.0 to 4.0.7

- `set global extra space for opt`
- `set global ipl pin limit`

Product Notes for Cadence PKS May 2001

- set global make routable max over congestion
- set global make routable max size
- set global make routable over congestion rate
- set global make routable oversize rate
- set global no of groute passes for cong
- set global pks do place option
- set global pks snap pin locations
- set global place over utilized
- set global smoothen area gap
- set global timing driven cong analysis
- set global use groute based cong analysis

For more details about the `set_global` commands, refer to the *Command Reference for Ambit[®] BuildGates[®] Synthesis and Cadence[®] PKS*.

Updated Commands

New in Version 4.0.8

There are no new updated PKS commands in version 4.0.8

Previously Included Updated Commands In Versions 4.0 to 4.0.7

`set_pin_location`

The `set_pin_location` command syntax changed in the 4.0 release:

```
set_pin_location pin_name [-new {[-side side] [-index number]} | {[-location location] [-layer layername] [-box {lx | ly | ux | uy }]}]
```

To maintain backward compatibility, the old syntax is still available for earlier releases:

```
set_pin_location pin_name [-place {left|right|top|bottom}] [-index number]
```

The `set_pin_location` command sets one of the following: pin side, pin index, pin layer, pin location, or pin geometry.

Product Notes for Cadence PKS May 2001

Note: The `-new` option must be used to use the new `set_pin_location` command syntax. If not, the syntax will default back to the old syntax (see above).

Either the side or the location for the pin is mandatory and both cannot be used simultaneously.

The `-new` option will be removed in a future release, as well as the old syntax for the command, and the user will have to migrate to the new `set_pin_location` syntax.

`set_pin_location` Options

`-new`

A mandatory option needed to utilize the new syntax. Will be removed in a future release.

`-side`

Defines the side that the pin is assigned to. Can be one of { `left` | `right` | `bottom` | `top` }.

`-index`

Defines the relative index for the pin on the side it has been assigned to. Any number greater than one is specified along with the `-side` option.

`-location {x y}`

The location for the pin. This cannot be used in conjunction with the `-side` option.

`-box {lx|ly|ux|uy}`

The geometry for the pin.

`-layer layername`

The layer on which pin needs to be placed.

Examples of the `set_pin_location` command

■ `set_pin_location clk left` (old syntax)

`set_pin_location clk -new -side left` (new syntax)

This command sets the location of the clock pin to the left side of the die and sets the relative order to be the last pin on the respective side.

■ `set_pin_location clk left 19` (old syntax)

Product Notes for Cadence PKS May 2001

`set_pin_location clk -new -side left -index 19 (new syntax)`

This command sets the location of the clock pin to the left side of the die and also sets the relative order to be the 19th pin on the left side.

- `set_pin_location clk -box {0 0 100 100}(new syntax)`

This command sets the pin geometry to be a rectangular box with origin $\{0, 0\}$ and height and width 100 relative to the pin location. This is only available using the new syntax.

- `set_pin_location clk -location {10000 20999} (new syntax)`

This command sets the pin location to be $(10000, 20999)$. This is only available using the new syntax.

- `set_pin_location clk -layer M1 (new syntax)`

This command sets the pin to be located in the layer with the name `M1`. This is only available using the new syntax.

create_placement_area

`create_placement_area [-name name] lx ly ux uy`

The `create_placement_area` command has been enhanced to include:

- Placement areas now work in conjunction with DEF rows.

Use the placement areas to limit the region where you want the cells to be placed. This is useful for:

- Low utilization design, such as forcing QP to a cluster in the middle.
- Part of a semi-hierarchical flow.
- Avoiding thin strips of row area (around macros).

**Product Notes for Cadence PKS
May 2001**

Fix List for Version 4.0.8

| PCR Number | TITLE |
|-------------------|---|
| 329061 | Need steiner Pessimism knob, other than layer RC multipliers |
| 338425 | Need a command to remove power rail |
| 348581 | <code>set_clock_tree_constraints</code> need stronger syntax check |
| 349376 | <code>do_route</code> shifts the pre-routes on rows with orientation FS |
| 351422 | <code>do_build_physical_tree</code> sets clocks to propagated mode. |
| 351851 | Crash if <code>number_stripes</code> not defined |
| 352257 | <code>write_layer_usages</code> puts extra copies of data into file |
| 354735 | CTPKS crashes with unrecoverable exception error |
| 354747 | TRACK creation problem in PKS |
| 356336 | How <code>read_def</code> handles floorplan infos |
| 356465 | Crash in <code>ctg_move_tree_repeater</code> s |
| 356548 | Core dump running <code>do_place</code> on design |
| 357465 | Fail in placement setup phase (Zero STEP in TRACKS) |
| 357501 | qp crash when number of allowable pins per side is exceeded |
| 359325 | <code>write_def</code> does not write all <code>shieldnet</code> attributes |
| 360558 | SubBreak runs for >10 hours |
| 361018 | Ultra Placer termination message during <code>do_route</code> |
| 362364 | PKS seg faults during drv fixing |
| 362598 | <code>set_net_phy_att</code> does not honor or warn on repeat of option |
| 363584 | Length field causes incorrect cap calculation |
| 363664 | Wrong behavior of <code>set_net_physical_attribute -clock_tree_ne</code> |
| 363690 | Antenna LEF handling tsmc 0.18um artisan parsing error |
| 364979 | Tool crashes on <code>do_xform_optimize_slack -pks</code> |

**Product Notes for Cadence PKS
May 2001**

Fix List for Version 4.0.7

| PCR Number | Title |
|-------------------|---|
| 339720 | do_optimize_and_place crashes on xxxx blocks |
| 339720 | do_optimize_and_place crashes on xxxx blocks |
| 344019 | do_optimize_and_place, block crashes with xxxx |
| 344019 | do_optimize_and_place, block crashes with xxxx |
| 346324 | Core dump while read_lef_update |
| 346324 | Core dump while read_lef_update |
| 354748 | PKS Segmentation Fault |
| 356064 | PKS crash when reading polygons in lef (Sun only) |
| 359158 | 4.0-s006 takes much longer runtime in congestion analysis |

Fix List for Version 4.0.6

| PCR Number | Title |
|-------------------|---|
| 343684 | Name mapping Problem between PKS/NSDBA |
| 349352 | dont_modify ignored when set on cellref instead of instance |
| 350382 | write_adb and read_adb crashes after do_xform_connect_scan |
| 350971 | Crash during do_extract_route_parasitic in qplaceApiClass |
| 352323 | Crash during PI |
| 352602 | Crash in get_timing command |
| 352631 | DB not cleaned after do_build_clock_tree failing |
| 352632 | Crash in do_build_clock_tree on monroe design |
| 353246 | Crash in utilization. No room in 0.03% utilized design |
| 353449 | TLF INSERTION_DELAY not taken into account by ctpks |
| 354331 | Qplace fails because of a COMPONENT PIN name issue |
| 354549 | do_extract_route_parasitics: QPLACE aborts |

Product Notes for Cadence PKS
May 2001

| | |
|--------|---|
| 354668 | Warning messages during do_route |
| 354919 | FMR in PDMCluster: :~PMDCluster |
| 357171 | PHY-400 messages when doing congestion analysis |
| 357657 | Skew Results from CTPKS not good |

Fix List for Version 4.0.5

| PCR Number | Title |
|------------|---|
| 344598 | A parse error occurs when the scan order file with bit-slice notation is being read into the database. |
| 345214 | When starting from RTL, PKS produces a lot of PHY-411 warning messages for generic components. |
| 347678 | PKS is not correctly locating IO ports on top of IO pads. |
| 349975 | PKS does not adjust slew for different thresholds in the libraries for the driver and load cells. |
| 350380 | <code>write_def</code> generates repeated NET properties and connections. |
| 351383 | PKS is transposing the XSTEP and YSTEP row values when <code>read_def</code> is followed by <code>write_def</code> . |
| 351933 | <code>report_timing</code> shows a large slack difference between version 4.0.3 and version 4.0.4. The negative slacks in version 4.0.4 causes insertion of an excessive amount of buffers that congest the design. |
| 352174 | The endpoint functions for reduced critical range for T1 may pick up wrong objects to optimize. For example, it may pick up non-critical fanins of instances in the critical range. |

Fix List for Version 4.0.4

| PCR Number | Title |
|------------|--|
| 338878 | <code>report_timing</code> crashes if DIEAREA information is missing from def. |
| 342242 | <code>do_route</code> ignores values from <code>set_port_capacitance</code> command. |
| 342720 | Some data is not re-initialized in PKS DEF writer. |
| 343669 | Free Memory Read (FRM) occurs when running <code>do_remove_design</code> . |
| 346076 | Hold fixing option is not working with <code>do_xform_timing_correction</code> |
| 346322 | Vertical rows cause issues including incorrect utilization. |
| 346445 | PKS partially deletes TRACK section while writing DEF. |

Product Notes for Cadence PKS May 2001

- 347124 PKS `read_def` does not work with hier delimiter.
- 347332 The standalone Qplace does provide users the option `placeSelectedCellNearIOonly` to place the specified cells (especially boundary scan cells) close to the I/Os.
- 347816 PKS is not able to fix hold time when critical range is specified.
- 347840 A `do_optimized adb` has a different slack after quit and re-read the same adb.